TASK -4

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>To-Do List</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<h1>My To-Do List</h1>

<ul id="task-list"></ul>

<form id="add-task-form">

<label for="new-task">New Task:</label>

<input type="text" id="new-task" name="new-task" required>

<button type="submit">Add Task</button>

</form>

<script src="script.js"></script>

</body>

</html>

body {

font-family: sans-serif;

}

#task-list {

list-style: none;

padding: 0;

margin: 0;

}

#task-list li {

margin-bottom: 10px;

}

#add-task-form {

display: flex;

margin-top: 20px;

}

const taskList = document.getElementById('task-list');

const addTaskForm = document.getElementById('add-task-form');

// Function to fetch tasks from backend (replace with actual implementation)

function getTasks() {

// Simulate fetching data

const tasks = [

{ title: 'Buy groceries' },

{ title: 'Finish report', completed: true }

];

return tasks;

}

// Function to display tasks in the list

function displayTasks(tasks) {

taskList.innerHTML = ''; // Clear existing list

tasks.forEach(task => {

const listItem = document.createElement('li');

listItem.textContent = task.title;

if (task.completed) {

listItem.classList.add('completed'); // Add class for styling

}

taskList.appendChild(listItem);

});

}

// Function to handle adding a new task (replace with AJAX/Fetch call)

function addTask(taskTitle) {

// Simulate adding task to backend

console.log(`Adding task: ${taskTitle}`);

getTasks(); // Simulate fetching updated data

}

// Initial display of tasks

displayTasks(getTasks());

addTaskForm.addEventListener('submit', function(event) {

event.preventDefault(); // Prevent default form submission

const newTaskTitle = document.getElementById('new-task').value;

addTask(newTaskTitle);

document.getElementById('new-task').value = ''; // Clear input field

});

// Import necessary libraries

public class TaskServlet extends HttpServlet {

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// Handle GET requests (e.g., retrieving task list)

List<Task> tasks = getTasks(); // Replace with actual data retrieval logic

response.setContentType("application/json");

response.getWriter().write(convertTasksToJson(tasks)); // Convert tasks to JSON

}

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// Handle POST requests (e.g., adding a new task)

String newTaskTitle = request.getParameter("new-task");

addTask(newTaskTitle); // Replace with actual task creation logic

response.setStatus(HttpServletResponse.SC\_CREATED); // Set created status code

}

// Methods for getting tasks, adding tasks, converting to JSON (replace with actual implementations)

private List<Task> getTasks() {

// ...

}

private void addTask(String title) {

// ...

}

private String convertTasksToJson(List<Task> tasks) {

// ...

}

}